HW 3: Neural Networks
Advanced Machine Learning

Prof: Yannet Interian

Due: February 16th 2017

For this homework you will submit write up with your solutions to Canvas. Your code should be written using the deep learning library Keras. I am providing some sample code.

1 Back-propagation equations (10 points)

Derive the back-propagation equations for binary classification problem using the cross-entropy loss function. Use the Relu ($max(0, x)$) as the activation function.

2 Neural networks for handwritten digit recognition (25 points)

Neural networks are among the most powerful machine learning models, which can represent complex, nonlinear functions. However, due to a high number of hyperparameters, neural networks are extremely difficult to tune for a particular task. In this exercise, we will apply neural networks for the task of handwritten digit recognition.

Data: you are given samples of 28x28 pixel images containing a handwritten digit with value 0-255. Our goal is to predict the digit 0 to 9 being drawn. Specifically, for this assignment, we will examine how the choice of different parameters affect the performance of a model. Find here `http://yann.lecun.com/exdb/mnist/` some results on this dataset.

1. (5 points) The sample code contains the code required to train a neural network. You will now train a 2 layers neural network. In the input layer each pixel maps to an input. The hidden layer has M=300 neurons. The output layer output maps a one hot encoding. We are using Relu as the activation of the hidden layer and softmax for the output layer. Use the training and validation set to figure out a good learning rate. Report validation accuracy for the following values of learning rate 1, 0.1, 0.01, 0.001, 0.0001, 0.00001. Interpolate between the best two values. Train for nb_epoch=10. (You need to recompile to restart the initial weights.)

2. (5 points) Now will examine how the size of a hidden layer affect the model's performance. Use demo code to train the model with hidden layer size parameters of 10, 50, 100, 300, 1000, 2000. Use learning rate 0.01 and train for nb_epoch=10. Report the testing error achieved by each of these models. Which hidden layer size parameter achieves the best performance? Are some of these model overfitting? How can you tell that?

3. (5 points) Now let's train a neural network that employs L2 weight decay and compare it with the network without regularization. Take $M = 300$ and $lr = 0.01$ and train a model for 10 epochs. Then lower the $lr = 0.001$ and train the model for another 10 epochs. Report the training and validation accuracy. What do you notice? Then train a model with L2 regularization with weight 0.002, $lr = 0.001$ for 30 epochs. Report your results.

4. (5 points) Dropout is a very popular technique in deep learning community that enforces regularization in the neural network. The basic idea behind dropout is simple: during the training a selected fraction of neurons are zeroed out. We will now explore how the choice of a dropout parameter affect the performance of the model. Use demo code to train the model with various dropout parameters ranging for 0 to 1. Take $M = 300$ and $lr = 0.001$ and 30 epochs. Report the testing and training error achieved by each of these models. Which parameter value achieves the best performance? Why? Does the dropout help to reduce testing error compared to the model without a dropout? Compare results with L2 regularization. You can experiment with lowering the learning rate after 10 epochs.

5. (5 points) Explore building a 3-layer network. Experiment with dif-

ferent parameters and regularization. Add your code to the write up and summarize your findings. For some ideas look at these results: `http://yann.lecun.com/exdb/mnist/`